

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

X-715-71-451

PREPRINT

NASA TM X- 65785

# THE ADVANCED ON-BOARD PROCESSOR - AOP

RAYMOND G. HARTENSTEIN  
CHARLES E. TREVATHAN  
WILLIAM N. STEWART

OCTOBER 1971



**GODDARD SPACE FLIGHT CENTER**  
**GREENBELT, MARYLAND**

(NASA-TN-X-65785) THE ADVANCED ON-BOARD  
PROCESSOR (ACP) R.G. Hartenstein, et al  
(NASA) Oct. 1971 28 p CSCL 09B

N72-15175

Unclas  
G3/08 13350

**THE ADVANCED ON-BOARD PROCESSOR - AOP**

**Raymond G. Hartenstein  
Charles E. Trevathan  
William N. Stewart**

**October 1971**

**GODDARD SPACE FLIGHT CENTER  
Greenbelt, Maryland**

CONTENTS

	<u>Page</u>
I. INTRODUCTION . . . . .	1
II. HARDWARE DESCRIPTION . . . . .	2
A. General . . . . .	2
B. Processor Module . . . . .	5
C. Memory Module . . . . .	12
D. Special I/O . . . . .	14
E. AOP Implementation . . . . .	14
III. SUPPORT SOFTWARE . . . . .	15
IV. RELIABILITY . . . . .	17
A. Component Level . . . . .	17
B. System Level . . . . .	17
V. APPLICATION CONSIDERATIONS . . . . .	21
REFERENCES . . . . .	25

ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	AOP System Block Diagram . . . . .	3
2	Processor Module Block Diagram . . . . .	6
3	Typical AOP Stacked Configuration . . . . .	16
4	Reliability Models . . . . .	19
5	Reliability Curves . . . . .	20
6	Typical Centralized Computer Configuration . . . . .	22

## TABLES

<u>Table</u>	<u>Page</u>
I    General Characteristics of the Advanced On-Board Processor (AOP) . . . . .	4
IIA   AOP Instruction Set - Minor OP Codes (Non-Memory Reference) . . . . .	8
IIB   AOP Instruction Set - Major OP Codes (Memory Reference) . . . . .	9
III   Memory Features . . . . .	13
IV   List of On-Board Computer Functions . . . . .	24

## THE ADVANCED ON-BOARD PROCESSOR - AOP

### I. INTRODUCTION

The goal of the Advanced On-Board Processor (AOP) development program is to design, build, and flight qualify a highly reliable, moderately priced, digital computer for application on a variety of spacecraft. Included in this development program is the preparation of a complete support software package which consists of an assembler, simulator, loader, system diagnostic, operational executive, and many useful subroutines. The AOP hardware/software system is an extension of the On-Board Processor (OBP)<sup>1,2</sup> which was developed for general purpose use on earth orbiting spacecraft with its initial application being on board the fourth Orbiting Astronomical Observatory (OAO-C)<sup>3</sup>. Although the OBP possesses the significant features that are required for space application, the OAO-C version uses discrete integrated circuits and core memory and consequently is rather large. When operating at 100% duty cycle the OBP is too power consuming for use on many smaller spacecraft. A major purpose of the AOP development program is to achieve significant improvement in both these areas. Computer volume will be minimized by implementing the processor and input/output portions of the machine with large scale integrated circuits. Power consumption will be reduced through the use of plated wire and, in some cases, semiconductor memory elements.

In review of the initial computer development effort, the design objectives were high speed arithmetic and logic capability, a flexible input/output design, a powerful interrupt philosophy, and an inherently reliable architecture. While the first three of these features are important for application in a real-time, time-shared operation, particular emphasis has been placed on optimizing computer reliability. Catastrophic single point failure possibilities must be avoided if the computer is to be trusted with performing functions such as delayed command execution, data dependent command generation, telemetry format control, spacecraft status monitoring and summary reporting, attitude control, power and thermal subsystem control, and experiment data processing and control. It is now generally accepted that one of the best means of achieving a high probability of success is through partitioning the computer into functional modules and then configuring the system with standby spare modules which may be individually activated by command from ground. This standby spare approach, coupled with a dynamic hardware memory protection feature and the ability to arbitrarily reassign and reprogram any memory bank, provides the high degree of reliability desired for the AOP system.

This document is intended primarily as an introduction to the AOP development program and minute detail of logic operation is not presented; however, sufficient information is included to fill the need of system level designers. A description of the AOP hardware features is presented in the next section of the report. This is followed by sections discussing support software, system reliability, and application considerations.

## II. HARDWARE DESCRIPTION

### A. General

The architecture of the AOP is based on a modular approach to provide for standby spare redundancy. As shown in Figure 1 the system is composed of two module types: a Processor Module which performs the instruction execution and input/output functions and a basic Memory module which is 4096 words in size. These modules are interconnected by means of dual redundant "data" buses which provide the parallel data paths and control lines necessary for data transfers between storage and the processor unit. If more than one processor module are flown, one would be in the active mode while the spare(s) would be in a standby (unpowered) state. Normally all memory modules would be in the powered state since internal cycle by cycle power switching, as discussed in the memory section, results in power being consumed only by the unit addressed on a given cycle.

The nominal AOP system consists of two processor modules and a minimum of two memory modules connected together by two data buses labeled A and B in the figure. As shown, the processors are "doubled ported" to address either bus as required while each memory is dedicated to a single bus.

Power converters, although not shown on Figure 1, are considered an integral part of the overall AOP system. The power converter is sufficiently reliable so that in many cases one unit will be adequate; however, providing one or more backup converters is certainly feasible. Since each application represents a sufficiently unique power converter situation, no general or nominal configuration is suggested here.

General characteristics of the AOP are presented in Table I. The total system shown under Physical Characteristics assumes a minimum configuration of one processor, one memory, and one power converter.

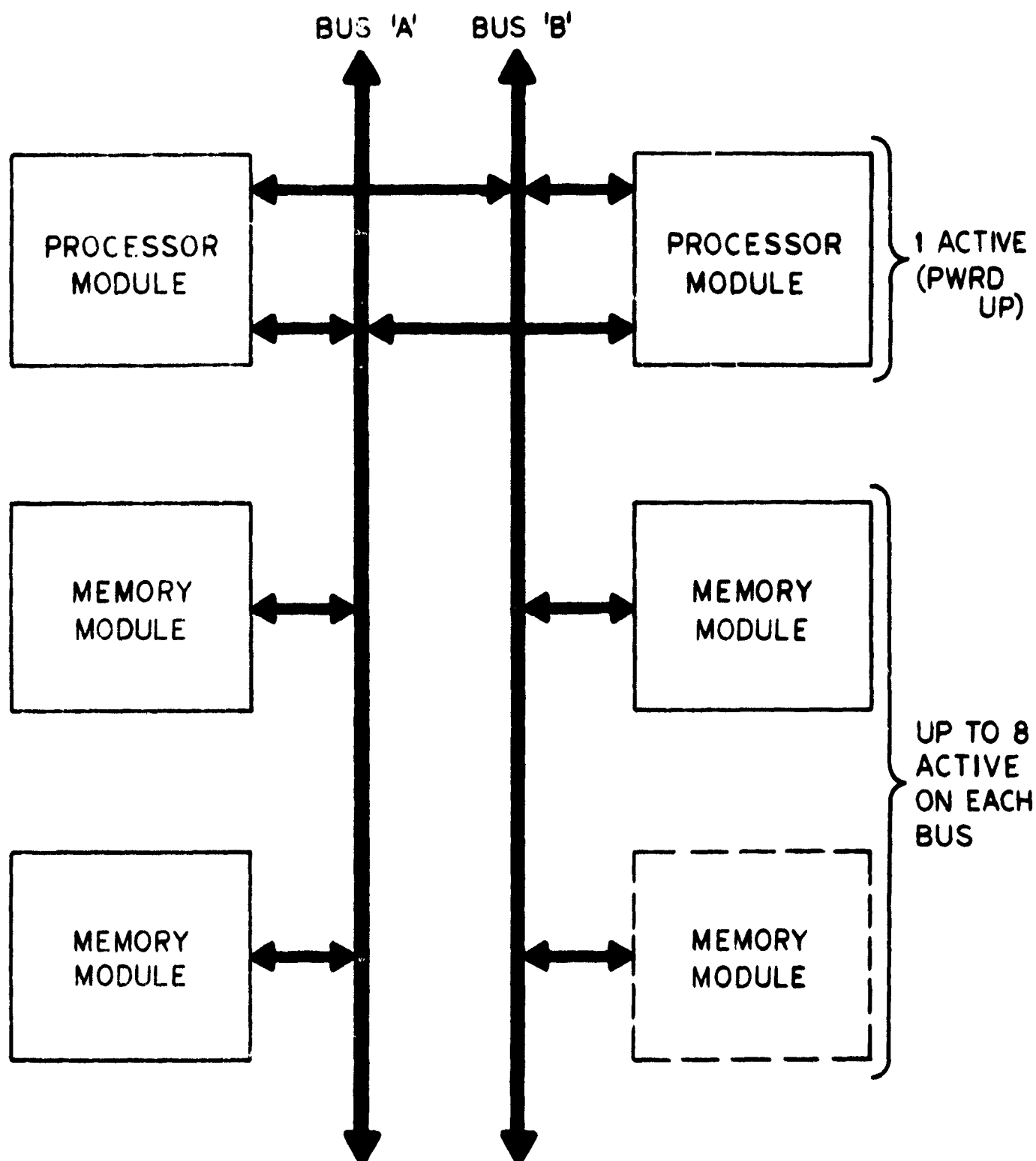


Figure 1. AOP System Block Diagram



**Table I**

**General Characteristics of the Advanced On-Board Processor (AOP)**

1. Low Power - 15 watts maximum while computing (including power converter and full memory complement) - 9 watts when halted.
2. High Speed - 4 microsecond Add, 30 microsecond Multiply, 60 microsecond Divide.
3. Large Memory - Expandable in 4 K modules to 64 K x 18 bit words. (Each 4 K module incorporates cycle by cycle power switching.)
4. 55 Instructions - 31 utilizing operand Fetch.
5. Simple to Program: (a) one double length accumulator (36 bits)  
(b) one index register  
(c) smooth handling of interrupts by hardware  
(d) powerful bit manipulating instructions  
(e) direct addressing of all 4096 words in any page.
6. Multilevel Interrupts - 16 priority interrupt levels with program control over lock-out status and interrupt disable.
7. Direct Memory Access (DMA) - Up to 16 independent cycle steal operations time share one DMA channel. Maximum I/O rate of 100 K words per second.
8. Memory Write Protection - In orbit reprogramming dictates that storage limits must be modifiable. Storage areas can be assigned in increments of 128 word blocks.
9. System utilizes a bus concept so that unpowered spare memory modules or processors may be flown. An automatic switch over control for real-time repair can be readily implemented.
10. Any spare memory bank may be used to functionally replace any other bank. The memory bank used for fixed (Interrupt and I/O) locations can be reassigned by command.

**Physical Characteristics**

	<u>Size</u>	<u>Weight</u>	<u>Power (Standby)</u>	<u>Power (Full Operation)</u>	<u>Technology</u>
Processor	75 in <sup>3</sup>	4 lbs.	5 watts	5 watts	TTL-LSI
Memory (4 K x 18)	100 in <sup>3</sup>	5 lbs.	0.07 watts	4 watts*	Plated Wire
Power Converter	100 in <sup>3</sup>	5 lbs.	3 watts	5 watts	Discrete
<b>Total (4 K) System</b>	<b>275 in</b>	<b>14 lbs.</b>	<b>8 watts</b>	<b>14 watts</b>	

\*Memory dissipation is rate dependent since the memory units are cycle by cycle power switched. Standby dissipation is 0.07 watts per 4 K unit hence the absolute maximum dissipation is 4 watts + 0.07 X number of memories or 5.12 watts worst case for a 64 K word memory.

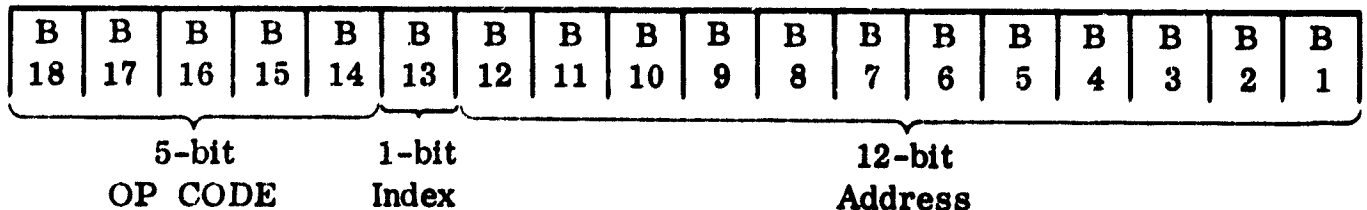
## B. Processor Module

(1) **Design Detail.** The Processor Module is a combination of a central processor and an Input/Output unit. In the original OBP system, these two units were separate packages; however, close scrutiny revealed that these two sub-systems are so closely related functionally, especially in the interrupt handling and I/O scheduling areas, that considerable hardware savings would result from their being more closely integrated. The penalty of having a more complex basic module was more than offset by these savings. Additionally, size, weight, and cost factors are reduced. It should also be pointed out that although the special interface circuitry (discussed later in this section) is unique to every application, it can be packaged in the processor box so that each processor unit is really a stand-alone functional unit requiring only memory. Figure 2 is a block diagram of the Processor Module.

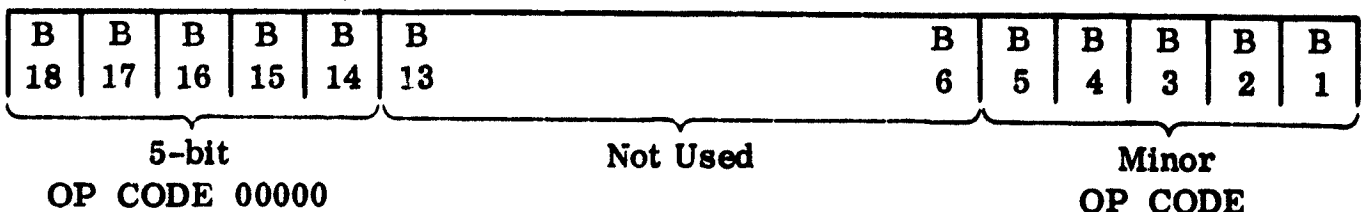
The Processor Module employs a fully parallel adder and parallel data transfers between registers and at the Processor/Memory interface. Data words and instructions are 18 bits in length with negative numbers being represented in two's complement form. Addressable hardware registers include an 18-bit index register, an 18-bit accumulator, an 18-bit extension of the accumulator, a 4-bit memory page register. In addition an 18-bit storage limit register which specifies read-only sections of memory and a 16-bit interrupt lockout register are conditionally addressable by means of privileged instructions. Unaddressable registers include a 16-bit memory address register, an 18-bit memory operand register, a 16-bit instruction counter, a 10-bit instruction register, and two 16-bit registers used for storing interrupt and cycle steal operation requests. Included for purposes of test and arithmetic control are three 1-bit registers: decision, carry, and overflow.

Processor Instructions are formatted as follows:

### MEMORY ACCESS



### NON-MEMORY ACCESS



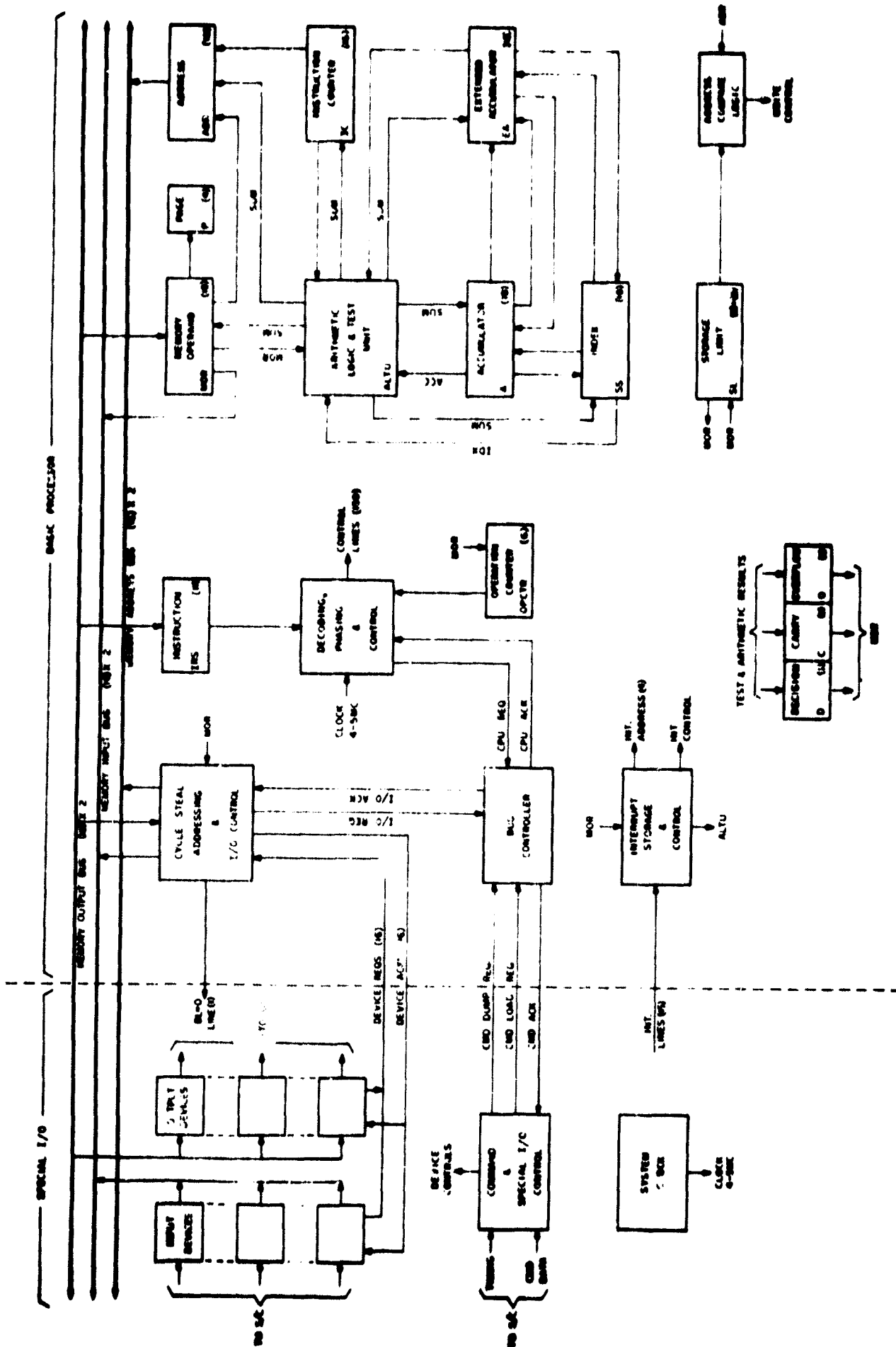


Figure 2. Processor Module Block Diagram

As shown in Table II, there are 55 instructions, 31 of which require an operand fetch. The other 24 instructions have a minor operation code in the address field of the instruction word. With 12 bits of address available, 4096 memory words are directly addressable. Memory size as large as 65,536 words requires a 4-bit page register which can be loaded and stored under program control and which is appended as the four high-order bits to the 12-bit address field to form a full 16-bit address. If the subscript bit (13) is set to a "1", the low order 16-bits of the subscript register are added to the address to form an effective address. All transfers are indirect such that pointers which are subject to change may be stored in data portions of memory. This obviates the need for write cycles in the protected program portions of memory.

(2) Special Features for Space Applications. There are 16 individually armed priority interrupts which allow asynchronous spacecraft events to gain access to computer operation at event dependent intervals. Although the 16 levels of interrupt are important to spacecraft design, the more significant aspect is the method of interrupt processing. The interrupt handling logic is designed so that when an interrupt is honored three critical registers are automatically saved and initialized to new values from fixed memory locations (completely independent of program operation). These are the Instruction Counter (sometimes called 'P' register), the Interrupt Lockout or "mask" register, and the Storage Limit register which allows the incoming program a limited area of memory in which it will be allowed to store data. The hardware handling of these registers is important in providing security of program execution in a long term unattended environment. Secure entry and exit from interrupts is assured since the fixed memory areas used for both "old" and "new" register values may be protected from "inadvertent" program modification (i.e., only the interrupt hardware is allowed to write in these areas). This is considered essential, especially in the long term unattended space environment where occasional memory errors, glitches, or transients must be considered a certainty rather than a remote possibility. The philosophy here is that the computer will be forgiven for an occasional wrong answer if it returns to output good answers. As a corollary to this, all critical decisions should be based on several right answers rather than one. Having the hardware automatically handle the mask and storage limit registers, additionally serves to simplify the ground rules under which the various programs must operate and assures, for instance, that when the executive program (exec.) is interrupted it will again regain control in spite of program bugs in the lower program areas. The exclusive use of an interrupt-like instruction (i.e., EXIT or RESUME) by the exec. as it branches out to "workers" extends all of these features downward to the worker programs — this can be important in a multi-programming situation where it is impossible to validate all programs in all modes of operation under all combinations of circumstances. The modification of the storage limit register provides for not only the protection of program

**Table IIA**  
**AOP Instruction Set**  
**Minor OP Codes (Non-Memory Reference)**

OP Code	Mnemonic	Time (Microseconds)	Description
000000	HLT	3	Halt
000001	TOV	3	Test Overflow
000002	NOP	3	Pass - No OP
000003	IGZ	3	Test Sign of ACC
000004	NEG	6	Negated
000005	IOP	22	Test ACC for Odd Parity
000006	ADC	4	Add Carry
000007	ROV	3	Reset Overflow
000010	CMP	6	One's Complement of ACC
000011	ISS	6	Increment if $\neq 0$ , if $= 0$ set D
000012	LPD	3	Load Page
000013	LDD	3	Push D into extension
000014	NORM	4*	Normalize ACC and EA
000015	IEA	6	Increment EA if $\neq 0$ , if $= 0$ set D
000016	EXIT	36	Transfer as on interrupt
000017	CPD	3	Complement D
000020	SIO	3	Set Int. Override
000021	IEZ	4	Test: ACC = 0
000022	FLP	3	Reverse ACC Bits
000023	RED	3	Reset D
000024	RIO	3	Reset Int. Override
000025	ACX	8	Exchange Acc. and Index
000026	AEA	8	Exchange Acc. and Extension
000027	EAX	8	Exchange Extension and Index

\*Add 1 microsecond for each place shifted.

**Table IIB**

**AOP Instruction Set (continued)**  
**Major OP Codes (Memory Reference)**

<b>OP Code</b>	<b>Mnemonic</b>	<b>Time (Microseconds) <sup>(1)</sup></b>	<b>Description</b>
02	ADX	4	Add to Index
04	ADD	4	Add
06	BRM	8	Branch and Mark Place
10	STE	6	Store Extension In
12	LDI	6	Load Indirect
14	SHF	5 (2)	Shift
16	OPT	6	Output To (Device)
20	LDA	4	Load ACC
22	XNGT	4	Test Index $\leq$ (Eff. Add)
24	SUB	4	Subtract
26	LT	4	Test ACC $<$ (Eff. Add)
30	ETR	4	Extract - Mask
32	STI	8	Store Indirect
34	CYC	5 (2)	Cycle
36	DSH	5 (2)	Double Shift
40	LDL	4	Load Operand Address

(1) For indexing, add 2 microseconds to all major Op Codes.

(2) Add 1 microsecond for each place shifted.

**Table IIB**

**AOP Instruction Set (continued)  
Major OP Codes (Memory Reference)**

<b>OP Code</b>	<b>Mnemonic</b>	<b>Time (Microseconds) (1)</b>	<b>Description</b>
42	BRC	4	Branch Conditional
44	MUL	24 +B(3)	Multiply
46	IET	4	Is Equal To
50	MRG	4	Ored With - Merge
52	LDE	4	Load Extension
54	LDX	4	Load Index
56	DCY	5 (2)	Double Cycle
60	STA	6	Store ACC
62	BRU	4	Branch (Unconditional)
64	DIV	58	Divide
66	IGT	4	Is Greater Than
70	EOR	4	Exclusive Or
72	TIN	22	Resume From ~ Terminate Int.
74	STX	6	Store Index
76	IPF	6	Input From (Device)

(1) For indexing, add 2 microseconds to all major Op Codes.

(2) Add 1 microsecond for each place shifted.

(3) B is number of ones in multiplier.

areas but also for the protection and therefore exclusive use of certain data areas. Thus, the area where delayed commands are stored by the executive program cannot be altered by any other program although other programs may utilize memory area nearby. In addition, the flexibility of dynamic storage protection allows for major program modifications from time to time while in orbit to accommodate new programs, program changes, data requirements changes, or memory unit failures which are also certain to occur at some time. This feature, however, does not preclude the possibility of using either hard wired memory lockout or read only memories for those extreme applications where permanent memory protection is necessary.

In addition to the asynchronous events which require program interruption, there is a great deal of data acquired and generated on board at various rates which should be transferred to and from computer memory without disturbing program execution. For example, data occurring at the command rate, telemetry rate, spacecraft spin rate, attitude control update rate, and experiment event dependent rate — all asynchronous relative to computer instruction execution rate — must be handled smoothly and efficiently. Program independent data transfers are accomplished by the AOP through the use of buffered I/O channels which operate in a cycle steal mode and time share a single set of Direct Memory Access (DMA) hardware.

Cycle steal addressing and control is one area of the I/O where considerable modifications of the OBP concepts have occurred. Both in order to expand the cycle steal capability and at the same time to considerably reduce the hardware required for this function a new approach of utilizing memory for I/O control was taken. Drawing on the interrupt technique, an area of memory is set aside to hold the necessary information relating to each of 16 independent cycle steal operations. Control logic (nearly identical to that used for interrupts) establishes priorities and selects which of the 16 devices is to be given control for the next input or output operation. If device 'n' requests a buffer operation and the request is granted a unique block length and storage address are retrieved from memory and the following sequence of events occurs: (1) the block length is retrieved from memory, (2) the block length is decremented and stored back in the memory, (3) the storage address is retrieved, (4) a data word is input or output to the indicated address, and (5) the storage address is incremented and stored back into memory. A CPU memory request (if present) must be honored before a second cycle steal operation is allowed to proceed. The cycle steal operation requires a total of 5 memory cycles giving a maximum burst rate of approximately 100 K words per second for a single channel operation. If the CPU was also busy this rate would drop to around 80 K words per second. It is important to note that the CPU (program execution) cannot be locked out by a faulty or busy I/O although it can be considerably slowed down (by perhaps 50%). This



is due to the slowness in getting requested memory cycles. On any cycle steal operation the block length may go to zero thus terminating the honoring of further requests for that device. In addition a special pulse is sent out to the I/O along with the last acknowledge signal so that a CPU interrupt may be set to signal the program that this buffer operation is complete. The program maintains complete control of all I/O operations by modifying the previously mentioned block length and storage address areas of memory and also may modify a mask register (ASR) for independently activating or deactivating any of the 16 I/O devices. The hardware for these 16 buffer control operations is an integral part of the basic processor unit; expansion to 32 or more buffer controlled channels can be accommodated.

A bus controller located in the processor module performs the function of scheduling the operation of memory since the I/O and CPU both require use of the memory data bus and only one may use it at a time. To maintain control in event of malfunction, both I/O and CPU may be pre-empted by the ground command link for purposes of memory loads or dumps. This capability need only be used when initializing or reloading the entire memory with program code. At other times, when normal computer operation exists, all input or output of data would occur under supervision of the computer itself, either through the I/O instructions (program control) or cycle steal buffer control.

### C. Memory Module

The random access memory modules for the AOP will be 4096 x 18 bit word plated wire memory units. The memories will feature cycle by cycle power switching, have a 600 nanosecond access time, and a 2 microsecond cycle time. They are non-volatile and read out is non-destructive. Table III summarizes these and other key features and includes, for comparison purposes, the corresponding characteristics of both the existing core memories to be flown on OAO-C and a proposed future semiconductor memory yet to be developed. As shown in the table, the major advantages of the plated wire versus the core memory is the decrease in power demand from 35 watts to 5 watts and the expected increase in MTBF. The latter is largely due to the fewer and lower power components of the plated wire units.

Cycle by cycle power switching reduces the power dissipated in the non-addressed memory units to 70 milliwatts which means that up to 16 memory units may be flown increasing the memory capacity to 64K words with only a moderate increase (1.2 watts) in the total power budget.

For future systems applications, several promising semiconductor memories (mostly of the C-MOS type) are being investigated. These would provide significant size and power improvements although, at present, lack of flight qualified circuits is the pacing factor. In addition, the fact that they are volatile

Table III

Memory Features

	Core Memory (as used on OAO-C)	Plated Wire Memory	(Proposed) Semiconductor Memory
Size	128 in <sup>3</sup>	100 in <sup>3</sup>	50 in <sup>3</sup>
Weight	6 lbs.	4 lbs.	2 lbs.
Power	0-35 watts (rate dependent)	0-5 watts (rate dependent)	C. 5 watt
Capacity	4096 words x 18 bits	4096 words x 18 bits	4096 words x 18 bits
Technology	20 mil cores 2 1/2 D stack Destructive Read Out	5 mil dia. plated wire 2 D stack Non-Destructive Read Out	C-MOS LSI
Access Time	800 nanoseconds	700 nanoseconds	600 nanoseconds
Cycle Time	2 microseconds	2 microseconds	1 microsecond
MTBF	40 K hours	90 K hours	100 K hours

(i.e., lose data on loss of power) must be considered in system design and might present limitations in some applications.

#### **D. Special I/O**

The mission's unique command decoding and all special interface logic required to establish communication between the AOP and the particular spacecraft subsystems including interface circuits are contained on a separate board called the Special I/O (Block Diagram shown in Figure 2). The processor timing can be provided either by a spacecraft clock input (in the vicinity of 4-5 MHz) or an internal oscillator which can be included in the Special I/O. The basic processor contains the logic to convert this signal to a two phase clock.

Since most data is transferred serially across the interface boundary a major task of the Special I/O is the conversion of data from the serial to the parallel (computer) format and vice versa. In addition, the portion of the hardware controlled memory load and dump capability which is peculiar to the command and telemetry formats of each mission is implemented in this special area. Inasmuch as the logic of the Special I/O has a considerable interface directly with the AOP memory data buses and I/O control logic, it is important that it be closely integrated with the basic processor and therefore, provision has been made for them both to be mounted in the same container. The interface lines can then simply be made by means of a short flex cable instead of requiring connectors and external cable runs.

Considerable emphasis is placed on the adequacy and adaptability of the Special I/O design because of its importance to mission success. Experience has shown that sufficient room for expansion must be provided in order to accommodate the constantly growing demands placed on the computer system as the spacecraft development proceeds from design to operational hardware.

#### **E. AOP Implementation**

Future mission requirements indicated that improvements in size, weight, power consumption and hardware cost of the OBP were required. The memory technology did not offer hope of significant reductions in this area in the immediate future except for the very expensive semiconductor types and these could only be used where their volatility could be tolerated. Therefore, a very hard look was taken at the CPU-I/O modules with regard to possible reductions since they represented about 50% of the hardware (cost and size). A two-pronged study was launched with the goal of reducing hardware by logic simplification and/or elimination along with evaluating the feasibility of converting from small scale integrated circuits to large scale integrated circuits. Several hardware modifications resulted which tailored the machine more to the problems actually

encountered on OAO-C. This produced a net reduction in hardware. The two major changes were the addition of two indirect addressing instructions and the simplification of multiply and divide operations to straightforward fractional arithmetic algorithms. Redesigning the I/O and combining it with the CPU, also resulted in significant hardware reductions.

The components study resulted in the selection of the newly developed Custom Metalized Multigate Array (CMMA). This is a Jet Propulsion Lab. sponsored program to develop a custom LSI chip containing 116 bi-polar gates. These devices are manufactured by Harris Semiconductor in accordance with JPL Specification CS-504841, dated January 20, 1971. The basic processor will be built using approximately 66 devices of 26 chip types. All logic will be performed by arrays of three input NAND logic gates which can be interconnected in any manner using a second layer of metallization. The crossing of signal lines is facilitated by pre-defined "cross unders" provided by the first layer of metallization which also performs the required interconnections of active devices to form the gates. Power connections to individual gates are made on the custom second level metallization so that unused gates may be left unpowered. The devices may be used like any bi-polar logic applying the familiar TTL logic rules with very few new considerations. The devices are mounted in standard 40 lead flat packages. The interconnect of the approximately 66 LSI packages will be performed by two multilayer boards approximately 5" x 9" each. These two boards will then be joined back to back and inter board connections made via plated-through holes. A connector or flex strip along a common edge will provide connections to the outside world (normally a special interface board will be mounted within the processor box). The card size and chassis have been influenced by the need to stack all units, memories, processors, and power converters, in some applications. A typical AOP stack is shown in Figure 3.

### III. SUPPORT SOFTWARE

The AOP support software package consists of an assembler, loader, and simulator. These programs are embedded in a flexible control language. The entire system is a link structured Fortran program with minimal use of assembly language. Hence the system is easily converted to machines other than the XDS 920. Minimum configuration is an XDS 920 with 16 K of core, 4 tape drives, card reader, and printer.

The assembler provides a full range of user services as well as a procedure capability. The assembler is most closely related to Meta symbol on XDS equipment or SLEUTH on Univac equipment.

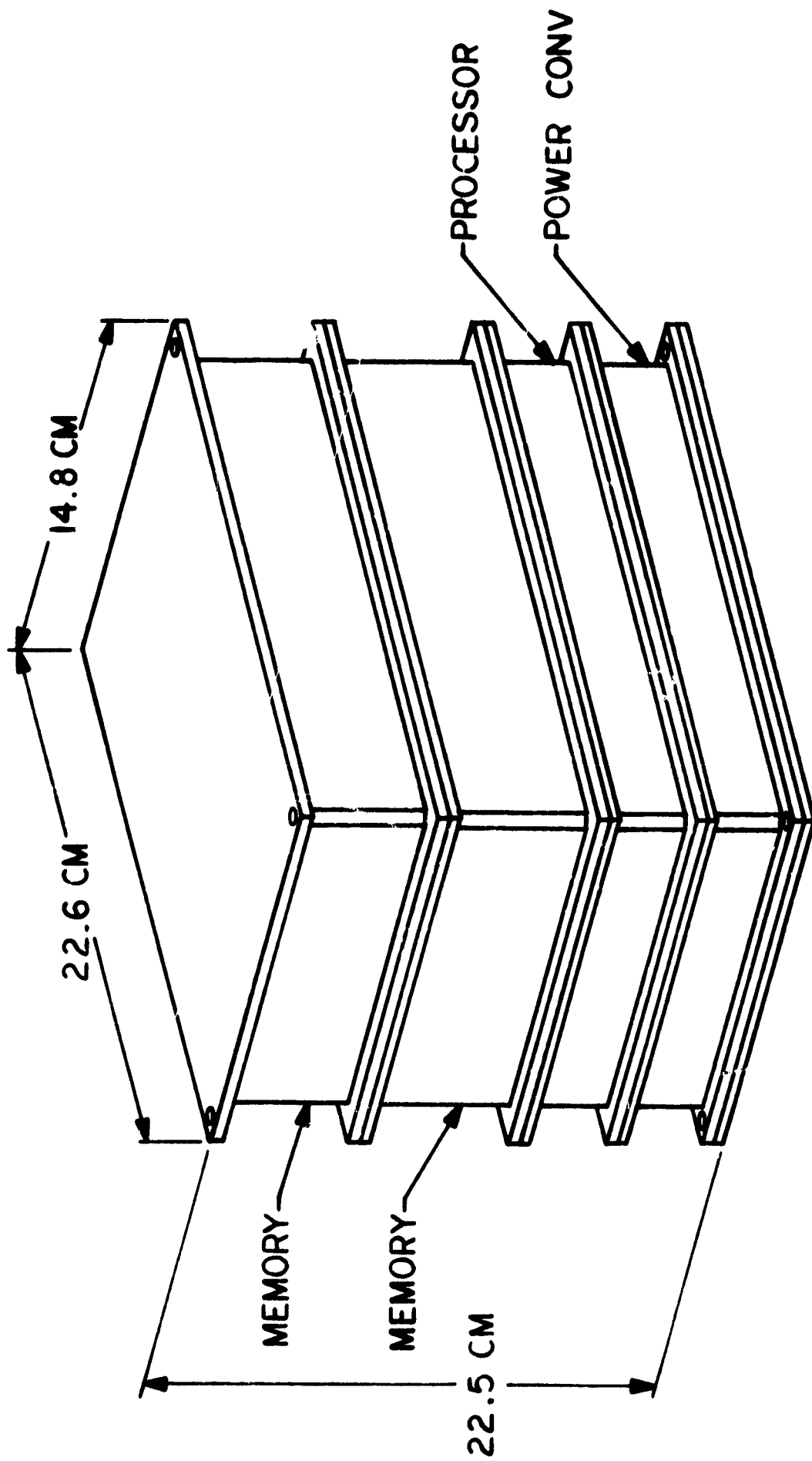


Figure 3. Typical AOP Stacked Configuration

The relocatable loader loads absolute/relocatable code or data linking the various specified modules and resolving external references. Although all code is packed together after loading, the Relocatable Loader has two modes of operation with respect to data: (1) data can be packed into consecutive locations, or (2) data for each module can be loaded in 128 word blocks to support the Test Executive storage protection feature. Output of the loader is an absolute core image tape suitable for loading directly into the AOP using appropriate Ground Support Equipment.

The simulator provides extensive timing and debugging facilities. The simulator package contains an interactive program which simulates the computer console to provide the instruction step function with register readouts. The simulation is slow (due mainly to the 8 microsecond cycle time of the XDS 920). Hence it is useful only for checking out programs in small increments and in cases where an AOP is not readily available.

#### **IV. RELIABILITY**

##### **A. Component Level**

The first concern in building a reliable system is to design it in the simplest possible manner utilizing the fewest possible components. Minimization of interconnections is also important but with the advent of LSI wherein most of the interconnection is done "on-the-chip" this factor is not nearly as significant as it once was. The reliability of the basic components, the building blocks of the system modules, is of paramount importance. The use of new LSI components in the AOP has introduced an unknown into the reliability equation. However, the impact of this is minimized by the fact that this component is built using well proven processes. Both the active devices (which have seen wide usage on a smaller scale) and two layer metallization technology (which has undergone extensive evaluation) have demonstrated a very high reliability. The combination of the two technologies into a larger geometry LSI device is proceeding on schedule. A demonstration run on the pilot production line has demonstrated a satisfactory yield of devices. The final phase in the development of the LSI components, the high reliability screening of several pilot production runs, is now in progress. The goal of this development program is to provide components for the Grand Tour (JPL) program requiring a ten-year mission life.

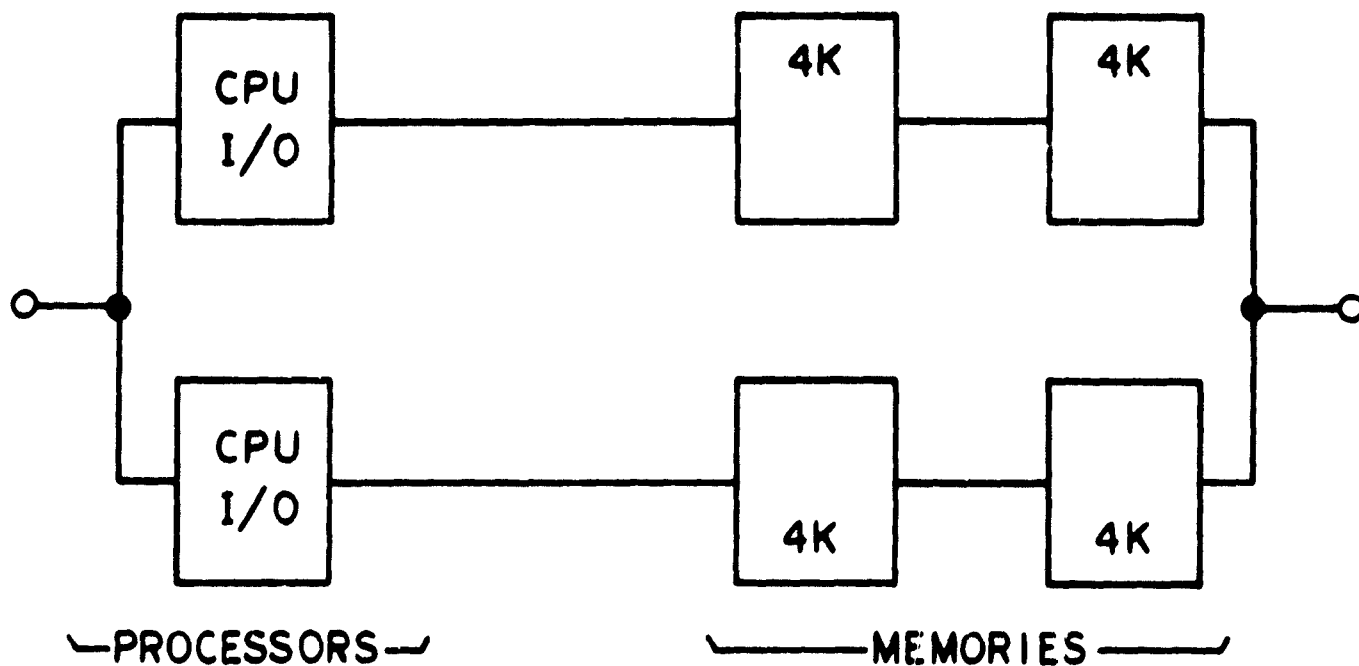
##### **B. System Level**

The approach taken to system reliability is largely dictated by the mission constraints. If the mission is of very short duration, such as a planetary fly by,

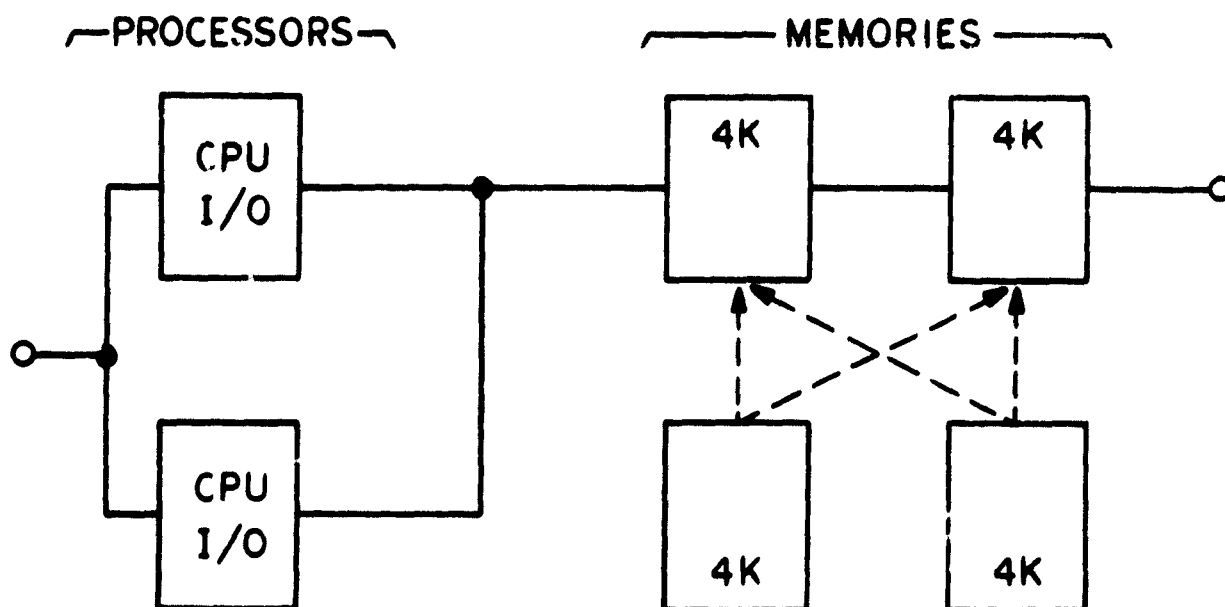
then the objective should be to get there with an operating system; reconfiguration of available components into a working system could be accomplished just prior to the event. Whether the reconfiguration occurs automatically or by ground operations may simply depend on whether ground contact exists. Another situation may require 100% computer operation for an extended period in which case a micro second switchover to a completely new standby system might be required — this dictates that a completely parallel software and hardware system be standing by and therefore eliminates the spare module approach. Spare modules might be included within each of the parallel systems but would not contribute to the prime operational concern. In this situation considerable attention must be given to the maintenance of the software in the standby unit and to the transient effects of the switchover on control systems and the like since it may take the new operating system time to become aware of spacecraft status.

For earth orbiting spacecraft the most common situation is the requirement to maintain system effectiveness for an extremely long period (five to ten years) with operational status being maintained by many reconfigurations of spacecraft equipments. It is then only necessary to maintain a safe standby condition while system reconfigurations occur usually under ground control. In all cases, a means external to the computer must be employed to evaluate its go/no-go condition and act accordingly. The usual criteria for such a check is to ask the computer "How are you feeling?" and if it either fails to answer or answers with anything but "I'm feeling fine" the assumption is that trouble exists and steps to correct the situation or at least remove the system from active status are taken. This is a reasonably safe assumption if the steps which the computer must take to arrive at its "I'm feeling fine" decision are exhaustive and conclusive.

At the system level the probability of mission success can be improved either by connecting two or more simplex systems in parallel (this is the configuration allowed by computer designs available today), or by adding redundant modules in parallel with the simplex systems modules to form a duplex system (this is the configuration allowed by the AOP architecture). The reliability models of each of these configurations are depicted in Figure 4. Assuming an application requiring 8 K words of memory and an equal mean time between failure (MTBF) for individual modules, relative success probabilities as a function of time have been calculated and are plotted in Figure 5. From these curves it can be seen that with the same amount of hardware the duplex (AOP) configuration is nearly twice as reliable as the dual simplex design. In addition, since the AOP memories have the power switching feature all four units of the duplex system including spares are available for use early in the mission. This effectively increases the memory size for non-critical tasks without cost to the system.



(a) Dual Simplex Design

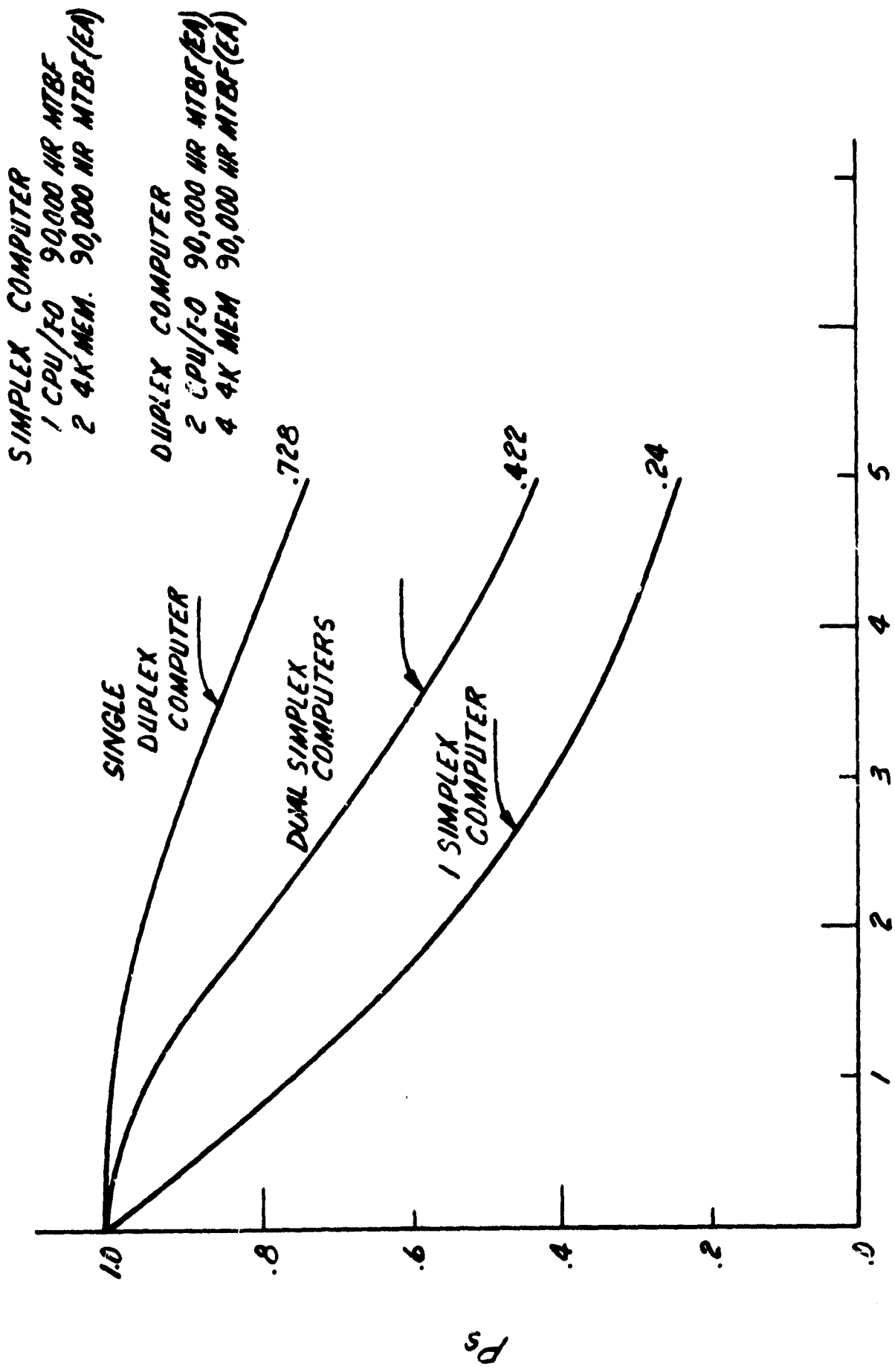


ANY 2 OF 4 MODULES MUST  
OPERATE-6 POSSIBLE COMBINATIONS

(b) Single Duplex Design (AOP)

Figure 4. Reliability Models





T IN YEARS  
 Figure 5. Reliability Curves

With the module/data bus approach any of the possible system configurations may be utilized depending upon the particular circumstances of each application. Several approaches to reliability of the AOP have been studied and are reported in detail in References 4 and 5.

## V. APPLICATION CONSIDERATIONS

The OAO-C mission will be the first in a series of applications of large scale on board computers to the solution of scientific spacecraft command and control problems. Few future satellites will fly without computers. The tremendous strides made in LSI and the reliability of computer logic and memories have made it difficult to justify the use of fixed logic even on the smaller satellites.

The programmability of the computer provides two overwhelming advantages over fixed logic: first, the amount of logic flown on the total spacecraft can usually be decreased several times over because the computer time shares the same logic for many tasks and, secondly, the way a task is performed or even whether it is performed can be easily modified at any time in the mission. This means that control functions or modes can be established after orbital experience is gained — this is especially important to new spacecraft. In addition, the tremendous capability for calculations and decisions opens up a whole new era of spacecraft operations wherein most of the day-to-day routine operations can be performed with very minimal ground support. Ground data processing can be reduced by two factors: the amount of experiment data gathered can be reduced by recording data only during significant events and/or according to complex algorithms and by doing data summarizing and other processing on board; in some instances these factors could obviate the need for a tape recorder by reducing the required storage capacity to within range of a large non-mechanical memory. In the area of special operations the computer can have many different command or experiment sequences which can be entered the instant a special event is recognized, whereas, in the past, only those events which could be precisely anticipated or which happened to coincide with a real time operations pass could be treated in a special manner.

To demonstrate how a computer may be interconnected on a spacecraft in a surprisingly simple manner and yet be adequately interfaced to perform all of these functions the following example is presented. As shown in the generalized block diagram of Figure 6, a typical system consists of a central computer that can communicate with the complete spacecraft by being connected only to the command and telemetry subsystems. Briefly described, the command input line is used for loading of computer program and delayed commands. The command output line allows the computer to execute delayed commands as a function of onboard events or spacecraft time. This output may also be used for transferring

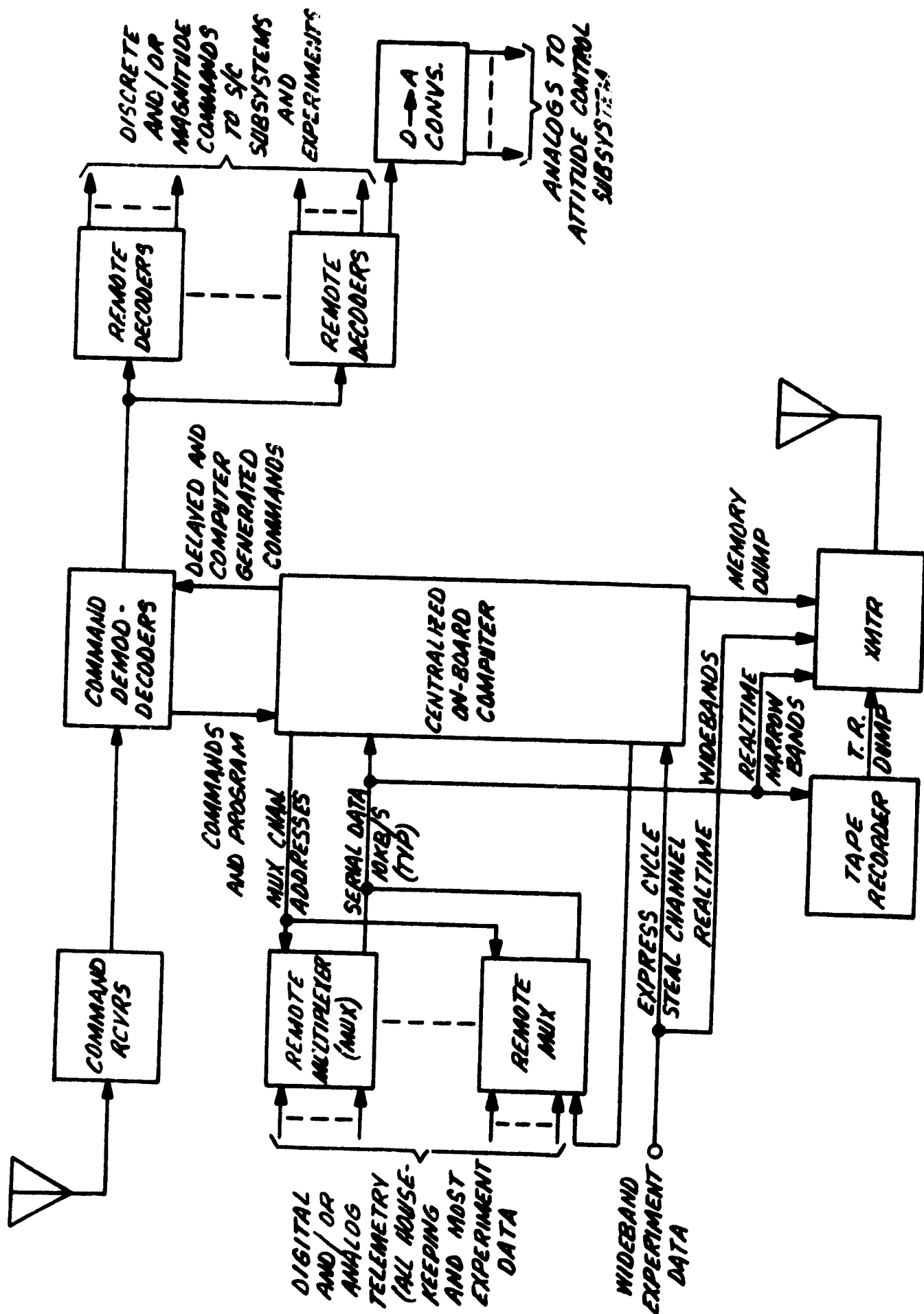


Figure 6. Typical Centralized Computer Configuration

computer generated control functions and attitude control error signals to appropriate subsystems via remote command decoders. Notice that analog control signals may be derived from digital-to-analog converters which can be rather arbitrarily placed at decoder outputs.

The interconnection with remote multiplexers of the telemetry subsystem allows the computer to not only have access to all onboard data but also to control the telemetry format. Generally there will be a baseline telemetry format controlled by multiplexer channel addresses derived from a Read Only Memory (ROM); however, computer generated formats would be ideal to optimize the telemetry system to varying experiment and spacecraft configurations which may be pre-planned or the result of onboard failures.

On the surface, the combined command and telemetry interface discussed allows the computer to serve as a stored command processor, a telemetry format controller, and a spacecraft data monitor. More significantly, however, this interface allows the computer to provide closed loop control of spacecraft attitude, power, temperature, and experiments.

Another possible input to a central computer may be wideband data from one or more experiments. In this case the computer may be required to either compress or process the data for later transmission to ground as well as use the data as additional basis for experiment control.

The last computer output shown on the block diagram is the line used for memory dump to the transmitter. This function is required for a variety of reasons: computer program verification, processed or compressed data dump, and summary message dump. Generation of summary messages could be one of the more important computer assignments in that it can provide a control center operator with a summary of current spacecraft status and a 'quick look' history of spacecraft activity since the last communication with the operator. Having access to this information early in the pass could be very valuable to both spacecraft maintenance and experiment scheduling.

Table IV attempts to summarize some of the many areas where the computer can contribute new or enhanced operations capability on future satellites. It should be emphasized that this is an incomplete list and in fact is growing everyday as new experience is gained on present day computerized spacecraft. One of the major lessons learned in early space efforts has been that spacecraft cannot remain fixed from mission to mission even within the same series. In fact complete reconfiguration on orbit could increase the mission return to the point of obviating additional missions. Built in flexibility via programmability is therefore an absolute must and for this requirement the computer is uniquely equipped to make an invaluable contribution.

**Table IV**

**List of On-Board Computer Functions**

**Command Handling**

1. Storage for Delayed Execution
  - a. Long Term Schedule
  - b. Special "Scratch Pad" Requests
2. Execution of Stored Sequences
3. Execution of Data Dependent Commands

**Data Handling**

1. Format Control
2. Data Compression
3. Status Summary
4. Operations Summary
5. Limit Checking

**Spacecraft Operation**

1. Emergency Control Sequences
2. Failure Workarounds
3. Thermal Control
4. Power Scheduling
5. Array/Regulator Control
6. Attitude Control
  - a. Stabilization
  - b. Pointing/Maneuvering
  - c. Momentum Management
7. System Test and Diagnosis

**Experiment Operation**

1. Event/Data Dependent Control/Commands
2. Routine Experiment Mode Control and Operation Scheduling
3. Test and Diagnosis
4. Sensor Stimulation and/or Calibration
5. Data Processing

## REFERENCES

1. "A General Purpose On-Board Processor for Spacecraft Application," Oct. 1968, C. Trevathan et al.
2. "Support Software for the Space Electronics Branch On-Board Processor," A. Merwarth and T. Taylor, X-562-68-388, Nov. 1968.
3. "An On-Board Processor for OAO-C," C. Trevathan, et al., A paper presented to ITC 1969 Conference.
4. "Reliability Techniques for Flight Systems," P. Heffner, X-560-70-323, April 1970.
5. "Reliability Study for OBP-CPU," Final Report for NAS 5-9753-28, May 1969 by Westinghouse Electric Corporation.